S4MPLE

User-guide

&

Technical Reference

InfoChimie
STRASBOURG

12-2012

Overview

1 Introduction

S4MPLE (Sampler For Multiple Protein-Ligand Entities) is a flexible molecular modeling tool, supporting empirical force field-driven conformational sampling and geometry optimization heuristics using a hybrid genetic algorithm (GA).

Allowing full control of the considered degrees of freedom (DoF), S4MPLE is a completely general approach to visit the conformational space of arbitrary molecules or molecular complexes. In theory, S4MPLE can be used in:

- conformational sampling of small organic compounds (1 entity),
- oligo-peptide folding (1 entity),
- protein loop repositioning in protein homology models (1 entity),
- ligand docking using various scenarios, from full rigid to partially flexible binding site (2 entities),
- simultaneous docking of several ligands into a same binding site (at least 3 entities).

By default, all atoms are mobile unless their rank numbers (in the order in which they are input) are listed in the ‟fixed_atoms” file (see §3.2.1). The list of explicit DoF, internally called fragments, to be used by the genetic algorithm is built automatically, based on chemical common sense (double bonds will not serve in torsion-drive mutations). Rings will be broken into fragments and subjected to intensive sampling only if a ring bond is listed in the ‟broken_bonds” file (see §3.2.3). Otherwise, ring geometry may only change due to external forces acting thereon during regular gradient-based minimizations

S4MPLE, written in object-Pascal, is used in command-line mode. As all molecular modeling approach, its main limitations are:

- the studied system size *vs.* available computational resources,
- the availability of force field (FF) parameters for the studied system.

This document facilitates, for the end-user, the installation and a first use of S4MPLE.

2 Installation procedure

S4MPLE has been compiled and tested on a linux x86_64 distribution (Mandriva 2010.2), and a shell is recommended to run it.

The latest version of S4MPLE (as binary file) is available from the laboratory website: http://infochim.u-strasbg.fr

After the extraction of the downloaded tar.gz file, you should have the following files and directories:

- Files
  - S4MPLE: binary file of the program
  - this user-guide in PDF format
- Directories:
  - parm_ff/: it contains the whole set of parameters for both available native FF (CVFF [1] or AMBER/GAFF [2, 3]). The latter FF is systematically used in any reported simulations.
  - parm_ff_fit/: it contains some additional parameters.
  - scripts_prepare_lig/: scripts needed to parameterize ligand molecule(s)
  - scripts_tuto/: scripts piloting the listed tutorials
  - tutorial_sampling/: an example to do conformational sampling of one ligand
  - tutorial_docking/: an example to do standard ligand docking
  - sup_data/: Supplementary material concerning redocking/benchmarking studies described in publications

2.1 Default setup

This default setup is the easiest way to install and run S4MPLE. It will works for projects which do not need the preparation of new ligands. Obviously, this default setup is sufficient to play with S4MPLE using the provided tutorials.

The default setup, step by step:

1) Move both parameter folders parm_ff and parm_ff_fit to some dedicated location on your file system..

2) Set (in your default .<whatever shell you use>rc setup file) the environment variable **SETUP_DIR** to point to the directory in which the parameter folders reside (so that 'ls –l $SETUP_DIR/parm_ff' will display the parameter folder contents)

3) Execute the following installation commands
   - cd $SETUP_DIR/parm_ff/
   - ./**set_amber99_gaff_core_ff.sh**
   - cd $SETUP_DIR/parm_ff_fit/
   - ./**set_amber99_gaff_fit_ff.sh**

4) For convenience, add the S4MPLE binary file to some directory in your **PATH**.

Now you should be able to launch both provided tutorials:

- Conformational sampling of an organic compound (see §5.1),

- Docking of a ligand into its binding site (see §5.2).

2.2 Full setup (involving AMBERTools)

In practice, the full setup is recommended for the end-user, since this procedure is needed in all cases where a new ligand is involved.

The AMBERTools software suite [4] is needed to prepare any new ligand. The release 1.3 of AMBERTools was initially used, and all tests were performed on this version.

AMBERTools can be downloaded, free of charges, at http://ambermd.org.

The full setup, step by step:

1) do the default setup (as described in §2.1)

2) install **JAVA**, available in any Linux distribution or from http://www.java.com

3) install the **ChemAxon** API, available from http://www.chemaxon.com

4) install a **Tcl** interpreter, available in any Linux distribution or from http://www.tcl.tk

5) install the **AMBERTools** software suite (see the provided installation file)

6) set the absolute path of the AMBERTools directory as the environment variable **AMBERHOME**
    - ls –l **$AMBERHOME/bin/antechamber $AMBERHOME/bin/parmchk** must not return any errors

7) For convenience, add the "scripts_prepare_lig" directory to your PATH

You should now be able to prepare new ligands according to the workflow described in details in §4.2

3 Technical reference

This chapter describes the technical points which are necessary for the end-user. It includes obviously file formats and available options. For more information about the genetic algorithm implementation or the FF-based energy function, the reader can refer to the associated papers[5, 6].

3.1 The input files and their formats

S4MPLE is able to use input files of various formats:

- .mol2: file in the SYBYL MOL2 format.
- .car: file in the CAR format originally introduced by BioSym InsightII (PDB-like with additional information like partial charges).
- .sdf: file in the MDL format.
- .mol: file in the MDL format.
- .out: this "chromosome file" contains poses or conformers encoded as a linear string. This kind of file is generated by S4MPLE and can be re-used to generate full-blown molecular files (one conformer per line).
- .fp: binary interaction fingerprint file. This kind of file is generated by S4MPLE and can be re-used to compute the fingerprint difference between that of the current conformer and that of a reference (alternative to RMSD criterion), or to guide conformational searching towards (attractor) or away (tabu) from conformations with interaction patterns similar to the one described in the .fp file

S4MPLE will seek in the working directory for input molecular structure files with the prefix "ref", in that order:

1) ref.mol2, ref.car: target file or protein-ligand complex file previously saved with S4MPLE,
2) ref.sdf, ref.mol: ligand file (mono or multi-structures).

It should be noted that at least one ref.xxx file must be present, and symbolic links can be used in practice to avoid renaming the input files.

Molecular output files will be, by default, generated using the input format. Their automatically generated names will include the current energy level and the pose rank, like in "best-30.8_0.mol2" – the pose of energy=-30.8 kcal/mol is ranked 0 (best pose found). Furthermore, additional information is stored in these files (in comment fields): the energy and its components, RMSD values with respect to input geometry of the ref file, *etc.*

3.2 Main configuration files

Various configuration files can be used, in addition to molecular files, to control the simulation. The most important ones are "fixed_atoms" and "hot_spots".

3.2.1 File fixed_atoms

This file is used to set the status (free or flexible) of all atoms in the system. It lists the fixed atoms in the system with 1 atom ID per line. The atom "ID" simply refers to the rank number of atom in the final list managed by the program. This follows the order in which atoms are read from input files, with assumed protein site files (of type .mol2 or .car) read first. Therefore, the bad news is that you need to perform some arithmetic if, for example, you wish to declare the second atom of the LIGAND as fixed, you need to add the atom ID N_{site}+2 in the fixed_atoms file, where N_{site} is the number of atoms in the receptor file. The good news is that you may block some ligand atoms, and free some protein moieties if you wish – classical docking/sampling tools will typically not allow this. By default (no fixed_atoms file), all atoms are considered flexible.

To perform a semi-flexible docking (free ligand but rigid binding site), just copy the atom lines from the site (mol2 file) in the fixed_atoms file.

Albeit S4MPLE does not need to make any assumption about the nature of the submitted molecules, in practice some of the docking options (hot spot definitions, use of geometric centers to direct ligand pose - see further on) do need to know which entity is the site, and which is the ligand. Please note that, by default, the site file is expected to contain at least one fixed atom – otherwise, S4MPLE will consider all the present entities as equivalent partners.

3.2.2 File hot_spots

This configuration file lists the hot spots for the initial ligand placement in the binding site. As for the fixed_atoms file, 1 atom ID per line must be provided. This gives a list of preferential anchoring points in the "active site" of the protein – it basically tells the tool where the "active site" is to be expected. Anchoring points must be either Hydrogen bond donors, acceptors or hydrophobic groups, which will be paired with potential partners from the ligand (in the ligand, all the H-bonding partners and hydrophobes count as "hot").

This file is useless in the context of a single entity: hot spot pairing serves to enable genetic operators to implicitly deal with inter-molecular degrees of freedom, implicitly controlling the position of the ligand with respect to the site. When a single molecular graph is subjected to sampling, covalent bonds are used to modify geometry.

If no hot_spots file are provided (not knowing where the active site is would be a good excuse), all qualifying (hydrogen bond acceptor/donor or carbons) site atoms are considered "hot". The latter case leads to full blind docking, but obviously needs a larger number of generations. This file can be automatically generated, if desired (see §3.3.5). Note that you need not specify ALL the potential anchoring points of the active site as hot_spots. This selection has no impact on force field energy calculations, it just pilots the initial attempt to randomly position the ligand with respect to the site. The algorithm will, for example, randomly draw hat spot #5, a hydrophobic carbon of the site. Then it will randomly pick one of the hydrophobic carbons of the ligand (if there are none, the tentative is aborted and a new site hot spot is picked). Say that the drawn lottery ball is carbon #7 of the ligand – then, the software will try to place the ligand in such a way as to establish a hydrophobic contact between hotspot 5 and carbon 7 (all while avoiding clashes, as far as possible). If, while trying to do this, the software also happens to bring together hydrogen bond donors and acceptors that are not listed as hot spots – very well, that will definitely count in energy evaluation. Therefore, it is recommended to list only the *most deeply buried* anchoring points of the site in hot_spots: the initial positioning of the ligand close to those will certainly cause other contacts to form spontaneously (if the correct pose does not imply interaction with the hot spots – that is not a problem, either: subsequent energy minimization may push the ligand away if needed: pushing the ligand out of the site is easy – dragging it in is difficult).

3.2.3 File broken_bonds

This file contains the bonds to break during the creation of the explicit DoF. One broken bond is defined by listing both involved atom IDs, separated by free spaces, in the bond (one definition per line).

Rings will be broken into DoF and subjected to intensive sampling only if a ring bond is listed in the "broken_bonds" file. Otherwise, ring geometry may only change due to external forces acting thereon during regular gradient-based minimizations. Automated recognition of rigid rings (that do not require any sampling) *vs* flexible rings that should be "opened", *i.e.* automatic generation of the "broken_bonds" file, is envisaged but not yet undertaken.

This option can be used to enhance the sampling of a flexible protein loop too.

3.2.4 File minfragsize

The minfragsize file is used to change the default behavior during the creation of the DoF. By default, one explicit "fragment" is created if one end contains a minimal number of atoms. Creating a "fragment" means that the program includes the explicit rotational degree of freedom associated to the fragment (the bond connecting it to the rest of the molecule) on the list of axes eligible for mutations/cross-overs. However, performing a cross-over swapping methyl groups is basically a waste of time. If an axis is not explicitly used for genetic operators because the fragment it carries is not big enough, this does not imply that its geometry is fixed – it may well change, following "Lamarckian" gradient-directed moves. It simply means that the program will never focus to explicitly alter the geometry of that specific moiety (also see below, with respect to the –n option of S4MPLE). It is possible to specify distinct behaviors between entities, as for example for ligands and flexible binding sites. Parameters are defined according to this convention (one per line): "<keyword> <value>". The available keywords are:

- mfs: set the same threshold for both ligand and target
- mfs_lig: set the threshold for ligand
- mfs_ target: set the threshold for target

For example, it is possible to use a low value (mfs_lig 3) for ligand in order to explicitly sample most DoF (moieties larger than a methyl group) while performing a fuzzier flexible site docking by using a larger threshold (mfs_target 6).

3.2.5 File runparams

This configuration file allows the user to change several critical parameters of a launched evolutionary-based simulation *after* the simulation has been launched. This file is revisited at each iteration, and internal updates are made on the fly. Parameters are defined according to this convention (one per line): ‟<keyword> <value>”. The available keywords are:

- ngen: change the number of desired generations
- runtime: change the current time limit

3.2.6 File active_pairs.lst

This file lists the atoms pairs to consider in the simulation. This file is usually created by the ‟prune” strategy of the evolutionary algorithm (see there), and can be useful in cases where a large binding site is defined with only specific flexible moieties. The latter strategy performs a preliminary simulation in order to prune some pairs from the whole non-bonded list (pairs involving atoms which are never close from each other are discarded). Using a pruning run before the actual simulation may significantly alleviate the pair list, but the user has to use it carefully.

3.2.7 File frozen_bonds

The file allows to constraint dihedral angles to their native values by listing involved atoms IDs as in the broken_bonds file (under testing at the moment).

3.2.8 File bonds

The final configuration file is only used in the context of the CAR format for the target. It lists explicitly the bonds, if they cannot be automatically detected (as in PDB file, there is no connectivity table for the usual protein residues in CAR file).

3.2.9 Chromosome I/O files

The term ‟conformer chromosome refers as a pose encoded as a linear string. This string contains all coordinates (which are converted to integers by multiplying by 1000, then rounding up) of the free atoms of the system. Fixed atom coordinates will always be equal to the initial values read from data files. Note that in a simulation in which you'd wish to gradually unlock increasingly large parts of the molecule (or, on the contrary, to fix moieties) you cannot use chromosomes obtained with a different fixed_atoms list. You will need to edit the chromosome file, and insert the input file coordinates for the freshly unlocked atoms in the string (at the correct positions), or, on the contrary, delete the coordinates of freshly fixed atoms.

Chromosome files (see pop_in and pop_out options below) are generated during the sampling process, in order to save the so-far near-optimal visited geometries. The first column of these plain text files will contain the force field energy value, followed by the chromosome string as defined above. They are reread for post-processing purposes and conversion to full-blown output molecular files of the best sampled geometries, for visualization in your favorite graphical interface.

3.3 Available options in S4MPLE

The command-line help is detailed here.

3.3.1 Main options

-h: use this option to print the command-line help.

-i <directory>: use this option to set the working directory. The latter must contain all molecular and control files.

-f <directory>: use this option to specify the force field directory relative to a reference location specified by the environment variable $SETUP_DIR (see §2.1). Thus, use –f parm_ff to use the native Core Amber/GAff, and –f parm_ff_fit to use the solvation-extended version.

The minimization of the system is called with the option -m, while the genetic algorithm is launched with the option -e (see below).

3.3.2 Specific management of DoF

-n <binary value>: use this option to specify the amide mode.
Available values are:

- 0 = amide bonds are not considered as explicit DoF
- 1 = amide bonds are explicitly sampled

The default value is 0 (FALSE).

-q <value>: use this option to automatically unlock polar hydrogens whatever their status from fixed_atoms file.
Supported values are:

- 0 = do nothing
- 1 = unlock hydrogens from hydroxyl groups
- 2 = unlock all polar hydrogens

The default value is 0.

3.3.3 Energy minimization switches

-a <value>: use this option to set the convergence threshold for the gradient.
The default value is 1.0e^{-7}.

-b <value>: use this option to set the required gain in energy to be achieved by a minimization step.
The default value is 0.10 kcal.

Within the full minimization procedure, a step not managing to lower energy by more than that counts as a ‟failure”, and minimization undertakes a bond softening/retightening cycle to escape the local minimum.

-m <value>: use this option to specify the maximal number of allowed ‟failures” in the above sense, before stopping the full minimization procedure.

3.3.4 Sampling and post-processing

-e <colon-separated options string>: use this option to specify the task to perform (evolutionary algorithm, post-processing) and pass the corresponding control parameters (see detailed discussion below).

-j <value>: use this option to specify the energy window width with respect to the so-far best energy. The default value is +30.0 kcal/mol.

-s <binary value>: use this option to set the output status.
Available values are:

- 0 = only save chromosomes (DoF of the system encoded as linear strings)
- 1= save molecules as both molecular files (one per instance) and chromosome

The default value is 1 (TRUE).

3.3.5 Miscellaneous controls

-y <binary value>: use this option to use geometric centers of entities to help the placement in the binding site when docking.
Available values are:

- 0 = Off
- 1 = On

The default value is 0 (FALSE). This assumes that the receptor file (ref.car or ref.mol2, in which at least one atom is declared fixed) only includes the active site neighborhood – i.e. represents a “sphere” of residues centered on the active site cleft (typically, we use MOE to cut a sphere of residues of 10 A around the active site), in order to make sure that the geometric mean of the site atom coordinates actually pinpoints the active site. Turning the geometric center option on triggers the program to double-check that the initial pose not only features some randomly picked site-ligand favorable contact, but also that the ligand is well located into the site (sometimes, if hot spots at the outer edge of the binding site are defined, S4MPLE may generate initial poses in which the ligand interacts with such a hot spot, all while pending outside the active site cleft).

-r <value>: use this option to set the RMSD mode.
Available values are:

- 0 = do not compute any RMSD
- 1 = compute standard RMSD
- 2 = same as 1 but using a symmetry-compliant RMSD for ligands

The default value is 2.

-o <value>: use this option to automatically create the hot_spots file.
If the geometric centers mode is on (see -y switch), then all putative hot spots (hydrogen bond donors, hydrogen bond acceptors and carbons) around the geometric center of the site are saved (maximal number = value).
If the geometric centers mode is off, then only the putative hot spots around the loaded ligand are extracted (maximal number = no limit).

-z <filename>: use this option to specify a custom reference binary fingerprint from an external file.
By default, the binary fingerprint of the starting geometry is employed.

-t <binary value>: use this option to set the superimposition mode for single-entity sampling.
Available values are:

- 0 = Off
- 1 = On

The default value is 0 (FALSE).

-w <value>: use this option to set the water contact mode.
Available values are:

- 1 = standard case, the water contact mode is off.
- number > 1 = the contact strength between ligand and water molecules is scaled by the value.

By default, the water contact mode is off (FALSE) and the scaling value is 1.0.

-k <value>: use this option to set the force constant used in site constrained minimization.

The latter are performed with respect to their native coordinates.

3.3.6 Sampler Switches

This sub-chapter sums up the currently supported task control options with the -e flag. The options have to be concatenated as option1=value1:option2=value2:... using colons as separators. The key option is the strategy, to be passed as ‟strat=chosen strategy keyword”.

Preliminary run

The prune strategy can be used to perform a preliminary simulation in order to prune some pairs from the whole non-bonded list (pairs involving atoms which are never close from each other are discarded). Using a pruning run before the actual simulation may significantly alleviate the pair list.

This will produce, locally, a list of non-bonded pairs that were found to be relevant. The file will be implicitly used by the following actual simulation, restricting the list of actually monitored NB pairs to a subset of these preselected pairs. The idea is that very far atoms may never directly meet, whereas they may indirectly interact by jointly exercising forces on intermediate geometric elements.

At the moment, this approach has not been extensively tested.

Running a GA-driven sampling

To actually perform a GA-driven sampling run, set option ‟strat” to one of the supported heuristics below (the default strategy is strat=evol):

- strat=evol: default evolutionary strategy,
- strat=base: basic GA,
- strat=elit: elitist GA,
- strat=tour: tournament-based GA,
- strat=hc: hill climbing GA.

The "evol", "base" and "hc" strategies have been extensively tested, the others less so. Note that evol requires many more generations, as it features a single individual which is modified by the operators per generation.

Further strategy-related options include:

- npop=<population size>
  This switch is used to set the population size. The default value is 50.

- ngen=<generation-number>
  This switch is used to set the number of generations. The default value is 500.
  Some few hundred will do for simple docking, but a full deployment on a grid will be needed for highly complex systems (work in progress).

- water=[true/false]
  This switch tries to perform an optimization of free waters around each kept pose using the GA during the post-processing stage. The default value is FALSE.

- minfpdiff=<value>
  This switch is used to set the minimal difference for interaction fingerprints (FP) of two non-redundant conformers (related to fingerprint size). The default value is 0.01.

- pop_out=<output filename>
  This switch is used to set the output filename in which conformer chromosomes are output.

- search=local
  Just add this option in the GA options string to force a sampling in the neighborhood of the input structure.

- seed=[true/false]

This switch is used to reset the random seed at the beginning of the simulation. Distinct simulations can be performed using this option. By default, there is no modification of the random seed.

- runtime=<value>

  This switch is used to set maximal allowed run time in hours for the simulation. The time limit is disabled (default behavior) when this parameter is equals to 0.

Post-processing stage

To post-process a brute conformer chromosome list, obtained from a previous simulation, use:

- strat=filter

  This strategy enables the post-processing stage. After reading the input conformer chromosomes file, poses are filtered according two criterions:

  - energy width,
  - redundancy using internal interaction fingerprints.

  All kept poses are directly saved or subjected to post-processing stage according to specified options.

- minfpdiff=<value>

  This switch is used to set the minimal difference for interaction fingerprints (FP) of two non-redundant conformers (related to fingerprint size). The default value is 0.01.

- pop_out=<output filename>

  This switch is used to set the output filename in which kept conformer chromosomes are output.

- pop_in=<input filename>

  This switch is used to set the input filename which contain the conformer chromosomes (poses as linear strings). This file is typically the output filename of the simulation.

- maxSaved=<value>

This switch is used to set maximal number of poses to save during the filtering stage. By default, this number is 200.

- optimize=<value>

  This switch is used to optimize the kept poses read from the input chromosomes file. It has the same behavior as with the -m option above, except that minimization is systematically applied to any conformer before saving. A value of 0 leads to the usual gradient-based procedure without softening/retightening cycles, while larger values enable the full minimization procedure (see -m option). The returned geometry will be confronted again with more stable ones, and if still not redundant it will be marked for saving.

- optoh=<value>

  This switch is used to enable the pre-optimization of hydroxyl groups.
  Available values are:
  - 0 = do not pre-optimize OH groups
  - 1 = pre-optimize hydroxyl groups from ligands
  - 2 = pre-optimize hydroxyl groups from full system

  The default value is 0.

- dark=<value>

  This switch is used to set the intrinsic bonus for darker fingerprints in kcal/mol per percent darkness (default=0) in order to preferentially select conformers with many realized interactions.

3.4 RMSD Calculation

S4MPLE can compute distinct RMSDs (Root-mean-square deviation). They are computed over all heavy atoms without superimposition (by default). Native coordinates are used as reference coordinates for RMSD calculation, but custom reference coordinates can be used using a specific field (<INIT>) in SDF/MOL molecular files (see §4.2.1)

Several RMSD are computed and stored in molecular files:

- All: standard RMSD over the full system

- Calpha: standard RMSD over C-alpha atoms of target
- Target: standard RMSD for the full target
- TargetFlex: standard RMSD over free atoms of target
- Lig_X: standard or symmetrical-compliant RMSD for the ligand X

4 Preparation of the molecular input files

This chapter is dedicated to the preparation of molecular input files, including organic compounds or biological entities.

4.1 Preparation of biomolecule files

These may be oligo-peptides or protein binding sites. They need to be prepared and saved as mol2 file, from their PDB entries with common graphical molecular modeling soft. Preparation includes reading the PDB file, fixing erroneous bond orders, assigning protonation status of acido-basic groups, checking whether force field parameter assignment is successful within your favorite modeling software (a necessary, but not always sufficient coherence test). All tests were performed with mol2 generated by the program MOE [7].

We recommend to cut out a sphere of residues within some 10 Å around the active site (in general – within the neighborhood of flexible moieties). Save this molecular subset to the .mol2 file. Make sure your modeling software does not try to overzealously fill in the empty valences at the cut points with hydrogens – that will artificially create $–NH_2$ and, perhaps, aldehyde groups in the final structure, confusing the AMBER FF parameterization process [which expects –NH-C-C(=O)- backbones].

When using the AMBER FF, the partial charges and atomic types are assigned, on the fly, from the customized topology file using topological indexes. Briefly, each atom from a given residue has an unique flag, and all usual residues have been processed in order to compute these reference flags.

The full list of allowed residue names with the available AMBER FF are:

• standard residues:	ALA ARG ASN ASP CYS GLN GLU GLY HIS ILE LEU LYS MET PHE PRO SER THR TRP TYR VAL
• usual patches:	ACE NHE NME AMN CXL
• water residue:	HOH

- special histidine: HID HIE HIP
- special protonation states: ASZ CYX GLZ LYZ
- cysteine in disulfide bond: CYM
- N-terminal residues (all): NALA NARG NASN NASP NCYS NCYX NGLN NGLU NGLY NHID NHIE NHIP NILE NLEU NLYS NMET NPHE NPRO NSER NTHR NTRP NTYR NVAL
- C-terminal residues (all): CALA CARG CASN CASP CCYS CCYX CGLN CGLU CGLY CHID CHIE CHIP CILE CLEU CLYS CMET CPHE CPRO CSER CTHR CTRP CTYR CVAL

Any other residues, such as co-factors, have to be prepared as ligands. At the moment, nucleic bases were not investigated. It should be noted that standard names can be systematically used for the 20 common residues. During the initialization of the program, special residues are discovered from their common name (e.g. HIP from a HIS in the site file, CALA from ALA, CYM from CYS ...) and the relevant partial charges and atomic types are assigned.

4.2 Preparation of a ligand

There are no AMBER dictionary files for ligands: GAFF must be used to obtain specific parameters, and partial charges also need to be provided by the user. The following ligand preparation workflow, consisting of several steps, is used in docking simulations:

- computing Gasteiger partial charges [8] using ChemAxon libraries [9] (property <CHG>),
- adding GAFF atomic types using the Antechamber tool [4] (property <TYP>),
- using the Parmchk tool [3] to check whether there are missing parameters (*e.g.* bonds, angles or torsions). In that case, Parmchk tries to compute the missing parameters using empirical rules, and the latter are added to their respective FF parameters files,
- generating a single conformer using ChemAxon libraries (avoiding starting from the expected solution) and saving initial coordinates from heavy atoms in property <INIT>.

4.2.1 Computing the partial charges

At the beginning of the project, it has been decided to use Gasteiger charges for ligands. Although the mol/sdf format is a good way to encode ligand information (connectivity table, information about formal charges and stereochemistry ...), there are no specific location for the partial charges. A special field, named CHG, is used to store these pre-computed charges.

The provided JAVA program **PrepLigFull**, based on the ChemAxon API (version 5.7), is routinely used to compute and store the partial charges, as desired in the ligand input files. Besides, this tool can be employed to predict other properties, among other the major microspecie at a given pH.

The command-line manual from PrepLigFull is displayed below:

```
Unix Shell > java -jar PrepLigFull.jar

PrepLig : currently supported command-line switches:

I/O
 -f : Input filename
 -o : Output filename
 -c : standardization string or file

Microspecies
 -fix_protons : do not modify protonation state of compounds
 -pH : pH for microspecies calculation (default=7.4)
 -min_ms_pop : Minimal threshold for microspecies (1 = 100%)
 -explicit_microspecies : Enumerate microspecies explicitly (see min_ms_pop option)

Tautomers
 -explicit_tautomers : Enumerate tautomers explicitly

Conformers
```

```
 -single : Only save the best conformer (lowest energy)
 -maxconfs : Maximal number of conformers
 -noH : Don't prehydrogenize the Conformer plugin
 -tlim : Time limit for Conformers calculation (in seconds for each molecule)

Other
 -save_coords : save initial coordinates in <INIT> property
 -h : Print this help text

Gasteiger charges are stored in the <CHG> property

Several options need valid ChemAxon licenses !
```

The simplest way to do this first preparation step is:

```
Unix Shell > java -jar PrepLigFull.jar -f input.sdf -o input_chg.sdf -fix_protons
```

If you are interested about microspecies prediction at usual pH:

```
Unix Shell > java -jar PrepLigFull.jar -f input.sdf -o input_chg.sdf -pH 7.4
```

If you want to save initial coordinates for subsequent use (e.g. computing RMSD in redocking experiment, but starting from a non-native conformer):

```
Unix Shell > java -jar PrepLigFull.jar -f input.sdf -o input_chg.sdf -save_coords -fix_protons
```

All these commands will add the property CHG into the output files. Mono or multi-structures can be used as input file, and a new conformer is generated from the starting structure.

4.2.2 Assigning the AMBER/GAFF atomic types

This step involves two key tools, developed by GAFF authors, namely Antechamber [4] and Parmchk. Several scripts were developed to perform the remaining steps into one single command.

- for a single ligand:

```
Unix Shell > gaff_00_mono.sh <input sdf file>
```

- for batch processing of a given directory containing ligand molecular files:

```
Unix Shell > gaff_00_all input <input directory>
```

5 Tutorials

Two distinct tutorials are provided to allow a quick test of S4MPLE. All simulation involves at least two main steps:

1) sampling or docking itself,
2) post-processing of previously saved conformers using gradient-based optimizations.

The ligand docking run includes a preliminary relaxation of the X-ray conformation of the ligand within the binding pocket.

5.1 Conformational Sampling

In this example, the small ligand serotonin is briefly sampled.

```
Unix shell: > ./ scripts_tuto/sampling_ligands.sh tutorial_sampling parm_ff_fit
```

The results are provided in the results.tar.gz files too.

All configuration files and options were previously described, respectively in §3.2 and §3.3.

5.2 Ligand docking

In this example, a small organic ligand is docked into its cognate binding site.

```
Unix shell: > ./scripts_tuto/docking_filtering_opt.sh tutorial_docking parm_ff_fit 1N1M
```

The results are provided in the results.tar.gz files too.

All configuration files and options were previously described, respectively in §3.2 and §3.3.

6 Bibliography

1. Hagler, A. t.; E, H.; S, L., Energy functions for peptides and proteins: I, Derivation of a consistent force field including the hydrogen bond from amide crystals. *J. Am. Chem. Soc.* **1974,** *96*, 5319-5327.

2. Pearlman, D. A.; Case, D. A.; Caldwell, J. W.; Ross, W. S.; Cheatham, T. E.; Debolt, S.; Ferguson, D.; Seibel, G.; Kollman, P., AMBER a package of computer-programs for applying molecular mechanics, normal-mode analysis, molecular-dynamics and free-energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications* **1995,** *91* (1-3), 1-41.

3. Wang, J. M.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A., Development and testing of a general amber force field. *Journal of Computational Chemistry* **2004,** *25* (9), 1157-1174.

4. Wang, J. M.; Wang, W.; Kollman, P. A.; Case, D. A., Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics & Modelling* **2006,** *25* (2), 247-260.

5. Hoffer, L.; Horvath, D., S4MPLE - Sampler For Multiple Protein-Ligand Entities: Simultaneous docking of several entities. *J Chem Inf Model* **2012,** *submitted.*

6. Hoffer, L.; Chira, C.; Marcou, G.; A., V.; Horvath, D., S4MPLE - Sampler For Multiple Protein-Ligand Entities: Methodology & Rigid-Site Docking Benchmarking. *J. Mol. Graph. Model.* **2012,** *submitted*.

7. MOE, Molecular Operating Environment. *http://www.chemcomp.com* **Chemical Computing Group Inc**.

8. Gasteiger, J.; Marsilli, M., A New Model for Calculating Atomic Charges in Molecules. *Tetrahedron Lett.* **1978**, 3181-3184.

9. ChemAxon Calculation of Partial Charge Distributions. http://www.chemaxon.com/marvin/help/calculations/charge.html (accessed Feb. 2009).